UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA

Estudo de vulnerabilidades da plataforma re:dy

Mestrado em Segurança Informática
Versão Pública

André Filipe Sobreira Garrido

Dissertação orientada por:
Prof. Dr.ª Dulce Domingos
Mestre Paulo Moniz

2017

UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA

Estudo de vulnerabilidades da plataforma re:dy

Mestrado em Segurança Informática
Versão Pública

André Filipe Sobreira Garrido

Dissertação orientada por:
Prof. Dr.ª Dulce Domingos
Mestre Paulo Moniz

Júri:

Vogais:

2017

AGRADECIMENTOS

A autoria, na capa, é de uma única pessoa, mas não tenho dúvidas de que sem o apoio de muitas outras esta tese nunca teria sido terminada.

Sendo difícil expressar o estímulo, a ajuda, a compreensão que cada uma destas pessoas representa, fica uma breve tentativa:

À minha esposa, Rita Ganhão, por todas as palavras de apoio e confiança. Acima de tudo, por todos os sacrifícios que teve de fazer para que a família se mantivesse unida e nada faltasse nos dias que não pude estar presente.

Ao meu filho, Rodrigo Garrido, pelos dias de saudades que deixei e pelos sorrisos sempre que regressava. Pelo que passou nos últimos meses, com muita coragem, percebendo que o pai tinha de estar longe! É o meu herói!

À minha orientadora, Professora Doutora Dulce Domingos, pela paciência, disponibilidade, orientação e exigência num trabalho que parecia não ter fim.

À EDP, pela oportunidade de me deixar estudar um tema que tanto desejava e ainda por me receberem na sua sede como se fosse um dos elementos da sua equipa.

Ao Mestre Paulo Moniz, pela disponibilidade e orientação numa tese proposta por mim.

Ao Professor Doutor Francisco Martins, pela amabilidade, simpatia e orientação em todos os momentos que não era obrigado a fazer.

Ao Sr.º José Faustino, colaborador do INESC, pela simpatia em me receber e pela sua opinião profissional relativamente à soldagem nas *plugs* re:dy.

A Carlos Mão de Ferro, colaborador da Fundação Champalimaud, pela amabilidade e apoio para que o *switch* re:dy fosse soldado.

Aos meus pais, Maria do Carmo Sobreira e Ulisses Garrido, pela vontade que demonstraram em que eu fizesse este curso. Nunca deixaram que nada faltasse.

Ao meu colega e amigo, Duarte Sousa, pela motivação extra dada através de novas ideias ou pelas palavras certas em horas muito «erradas» da noite.

Ao meu colega e amigo, Ivo Vacas, pelos momentos em que me fez acreditar que era possível, nunca me deixando desistir.

Aos meus colegas do Mestrado em Segurança Informática, pela união, amizade e motivação para sermos os melhores enquanto pessoas e profissionais. Muitos dos nossos nomes escreverão certamente o futuro na Segurança Informática em Portugal!

À FCT - Fundação para a Ciência e a Tecnologia, através do financiamento ao projeto DOIT, com a referência PTDC/EEI-ESS/5863/2014, e à Unidade de Investigação LASIGE, com a referência UID/CEC/00408/2013.

Resumo

O ZigBee é um protocolo de comunicação sem-fios com a particularidade de ter um consumo extremamente baixo, sendo cada vez mais usado em tecnologias dedicadas à *Internet of Things*. A segurança tem sido uma constante preocupação, pensada desde a primeira especificação da norma e melhorada nas versões seguintes.

Este trabalho tem como principal objetivo analisar a segurança do protocolo ZigBee. Foi efetuado um levantamento das vulnerabilidades do ZigBee e das ferramentas disponíveis para auxiliar a deteção destas vulnerabilidades.

Como caso de estudo é utilizada a plataforma re:dy, um sistema comercial que tem como objetivo obter informações detalhadas e controlar local ou remotamente as tomadas inteligentes de casa de um cliente EDP. Através deste produto é possível ligar ou desligar equipamentos, receber alertas em casos de anomalia, ter noção do consumo real ou num dado período de tempo, entre muitas outras funcionalidades.

[Texto removido por motivos de confidencialidade].

Ao nível das ferramentas para deteção de vulnerabilidade, foi detalhado o método de instalação do *firmware* KillerBee para a *pen* RZUSBstick através da programadora JTAG Atmel-ICE. Durante a realização deste trabalho foram também reportados erros de captura de pacotes que deram origem à terceira versão do *firmware* KillerBee, oficialmente testado pelo autor deste trabalho.

[Texto removido por motivos de confidencialidade].

Palavras-chave: ZigBee, segurança, deteção de vulnerabilidades, re:dy

Abstract

ZigBee is a wireless protocol with an extremely low energy consumption rate and has been widely adopted in Internet of Things technologies. A predominant concern for security has been a characteristic since its first standard and has remained in all the following versions.

The main goal of this work is to analyse ZigBee security. It includes a vulnerability survey and a list of all the available tools to help detecting those vulnerabilities. As a case-study, we use the re:dy platform, a commercial system with the purpose of obtaining electrical consumption detailed information and controlling locally or remotely smart plugs of EDP's clients home. Through this product it is possible to switch equipments on or off, receiving alerts for any anomalies, monitoring consumption in real-time or over period of time, among many other functionalities.

[Part of the text was removed due to confidenciality reasons].

Considering contributions for vulnerability detection tools, the method to install the KillerBee firmware for the RZUSBstick through the Atmel-ICE JTAG programmer is detailed. During this work, it was also reported packet capture errors, which gave rise to the third version of KillerBee firmware, officially tested by the author of this work.

[Part of the text was removed due to confidenciality reasons].

***Keywords*:** ZigBee, security, vulnerability detection, re:dy

Conteúdo

Lista de Figuras

Lista de Tabelas

1 Introdução

A *Internet of Things* (IoT) tem sido apontado como o próximo grande passo a nível das comunicações. Espera-se que até 2020 existam cerca de 20 mil milhões de equipamentos ligados à Internet [25], registando informação através de equipamentos para controlo de casas e cidades inteligentes, ou mesmo de carros e *wearables*.

Este aumento de equipamentos ligados à Internet é visto pelas empresas como uma nova forma de gerar lucros. Não só passam a vender os produtos, como passam a ter acesso aos hábitos do seu utilizador, algo já explorado por companhias como a Google, Twitter ou Facebook. [38]

Uma das grandes preocupações do crescimento galopante das IoT deve-se à possibilidade de equipamentos com informação pessoal e em número tão grande serem exploradas por agentes maliciosos. São já conhecidos vários casos de câmaras de monitorização de bebés exploradas por hackers, chegando a haver interação com as vítimas [37]. No entanto, a maior preocupação prende-se com a possibilidade destes equipamentos, com capacidades de computação consideráveis, continuarem a ser usados para ataques de *Distributed Denial of Service* [22]. O que já é uma realidade atualmente [4] terá um crescimento exponencial e assustador se pensarmos que haverá um crescimento de 12 milhões de equipamentos, passíveis de serem explorados, nos próximos três anos.

A grande particularidade destes equipamentos é que são geralmente muito pequenos e com necessidades de usos de bateria por longos períodos de tempo. Principalmente este último ponto, tornou difícil a implementação dos protocolos existentes, dando origem ao ZigBee.

O ZigBee caracteriza-se por ter baixo consumo de bateria e baixo custo de produção, usando sinais de rádio intermitentes de baixa frequência e de baixa largura de banda. Tentando não cometer erros históricos de outras tecnologias, a segurança foi uma das preocupações desde o início do protocolo, usando, por exemplo, mecanismos para garantir a confidencialidade e integridade das mensagens. No entanto, existem limitações óbvias na implementação deste tipo de medidas, devido ao poder de computação limitado, consumos exagerados de bateria ou necessidade dos equipamentos terem de ser fisicamente invioláveis.

O trabalho aqui apresentado visa analisar, estudar e descobrir vulnerabilidades de segurança no protocolo ZigBee.

1.1 Motivação

O ZigBee, sendo um protocolo sem fios de baixo consumo, de livre uso e já com uma grande aceitação no mercado, espera-se que venha a ser cada vez mais utilizado como tecnologia preferencial da IoT.

Tendo em conta que este protocolo é implementado, por exemplo, em «casas inteligentes» ou sistemas de saúde, o fator segurança torna-se muito importante. Além do perigo direto de ligar ou desligar um equipamento sem supervisão, torna-se preocupante se um atacante tiver acesso aos dados em claro que passam na rede. A privacidade dos utilizadores e a confidencialidade dos dados tem de ser um requisito essencial. Contudo, alguns estudos indicam que essa não é a realidade. [19][24][42][55]

A segurança tem sido uma preocupação desde a primeira especificação do protocolo, seja através do uso de algoritmos como o AES CCM* ou o uso de várias chaves de cifra para as diferentes camadas existentes. Apesar das intenções, as limitações computacionais restringem o uso de medidas mais adequadas, como por exemplo cifras assimétricas, obrigando a que os dispositivos sejam fisicamente invioláveis, de forma a armazenar as chaves simétricas de forma segura.

1.2 Objetivos

O principal objetivo deste trabalho é analisar as vulnerabilidades do protocolo ZigBee. Deste modo é efetuado um levantamento das vulnerabilidades do protocolo recorrendo à análise do trabalho relacionado e um estudo das ferramentas utilizadas para deteção de vulnerabilidades.

Como caso de estudo é utilizada a plataforma re:dy, um sistema comercial fornecido pela EDP, com o propósito de oferecer um plataforma de tomadas inteligentes, ao mesmo tempo que disponibiliza a informação de consumo ao cliente final. O cliente através do website re:dy pode ligar e desligar tomadas de vários tipos. Existem tomadas simples, que apenas permitem ligar e desligar, tomadas com emissão de infravermelhos para ligar equipamentos de ar-condicionado e outros componentes mais técnicos com diferentes propósitos. Tendo em conta que o sistema necessita de acesso à Internet, através de uma rede IP, o estudo será alargado para a descoberta de vulnerabilidades tanto na rede IP, como no *website*.

1.3 Contexto Institucional

Este trabalho foi efetuado no âmbito do Mestrado de Segurança Informática da Faculdade de Ciências de Lisboa, da Universidade de Lisboa, em colaboração com a EDP. Esta empresa cedeu os equipamentos necessários para os testes efetuados, tal como permitiu a utilização dos sistemas de pré-produção.

1.4 Contribuições

Durante a realização deste trabalho foi feito um levantamento das vulnerabilidades existentes para o protocolo ZigBee. Além das vulnerabilidades, foram ainda resumidas as

ferramentas necessárias para a exploração desses ataques, havendo um grande destaque para a framework KillerBee.

[Texto removido por motivos de confidencialidade].

Devido ao nível de maturidade das ferramentas existentes, foram reportados vários erros nos fóruns apropriados. Isto levou a que houvesse uma aproximação à comunidade de segurança responsável pelo KillerBee, especialmente a Ryan Speers.

Esta colaboração resulta em duas contribuições adicionais:

- Descrição do procedimento de instalação do *firmware* KillerBee na placa RZUSBstick com a programadora JTAG Atmel-ICE, que até ao momento nunca tinha sido usada para este propósito.
- Após a verificação da existência de vários erros na captura de pacotes, houve a necessidade de realizar testes de verificação da terceira versão do *firmware* KillerBee.

1.5 Organização do Documento

O documento apresentado divide-se em cinco partes. A primeira parte é o capítulo presente, onde são expostas as motivações, objetivos, contribuições e a forma como é organizado o documento.

No capítulo seguinte é explicado o protocolo ZigBee, dando ênfase às suas camadas, medidas de segurança existentes e é realizado um comparativo com outros protocolos com propósitos idênticos. No final desse capítulo são descritas quais as vulnerabilidades e ataques existentes e as ferramentas usadas para esse objetivo. Aqui apresenta-se o estado da arte.

Os capítulos seguintes materializam a metodologia escolhida.

O terceiro capítulo é o estudo das vulnerabilidades da plataforma re:dy, com ênfase no protocolo ZigBee. Neste, descreve-se a plataforma re:dy, sua arquitetura e equipamentos. Ainda é explicado o laboratório montado para a realização dos testes em ZigBee e identificados os testes realizados.

O quarto capítulo é centrado no estudo das vulnerabilidades na rede IP e no *website*, sendo descrito mais uma vez o laboratório usado, as vulnerabilidades existentes e os testes aplicados.

O último capítulo finaliza a tese concluindo com um resumo dos resultados dos testes efetuados e desenha-se o trabalho a ser desenvolvido no futuro.

2 ZigBee: a norma e vulnerabilidades

Neste capítulo é apresentado o ZigBee, através de uma breve explicação técnica e histórica, as camadas que o definem e as medidas de segurança implementadas. Por razões lógicas, será feita uma comparação entre o ZigBee e protocolos similares, como é o caso do Wi-Fi e do Bluetooth.

No final serão enumeradas as várias vulnerabilidades e ataques conhecidos, assim como as ferramentas utilizadas para os realizar.

2.1 Introdução

O ZigBee é um protocolo baseado na especificação IEEE 802.15.4 usado para comunicação sem-fios. O seu propósito é o baixo consumo de bateria e baixo custo de produção e implementação quando comparado com outras tecnologias, como o Wi-Fi ou Bluetooth. Com este protocolo é possível criar *Home Area Networks* (HAN), baseado em perfis dedicados à automação de casa, equipamentos médicos, entre muitas outras funcionalidades. Estas redes usam sinais de rádio intermitentes de baixa frequência e de baixa largura de banda, havendo, no entanto, a necessidade de proximidade entre equipamentos.

Apesar do sinal apenas atingir distâncias entre os 10 e 100 metros, é possível transmitir dados por longas distâncias devido ao uso de topologia em malha (explicado na secção 2.2.2). Além deste tipo de topologia, a camada de rede suporta ainda a topologia em estrela.

O ZigBee é composto por quatro camadas: física, *media access control*, rede e aplicação. Os equipamentos que permitem realizar as comunicações são: *ZigBee Trust-Center*, *ZigBee Coordinator*, *ZigBee Router e ZigBee End Device*.

Dependendo do continente onde se opera o ZigBee, a frequência pode ser diferente. Para o continente Europeu opera-se na frequência dos 868MHz, enquanto que para o continente americano está disponível a frequência 915MHz. O Japão usa os 950MHz e a China os 780MHz. Apesar das diferentes gamas de operação, é possível ser usada a frequência 2.4GHz a nível internacional.

A evolução deste protocolo tem sido constante desde a sua criação. Começou a ser usado em 1998 mas só em 2004 foi tornada pública a primeira especificação que dava pelo nome

ZigBee-2004. A base do protocolo já se encontrava nesta versão, mas a necessidade de incluir capacidades de endereçamento de grupos, passando a permitir realizar *multicast*, obrigou à ratificação do ZigBee-2006. Um ano mais tarde foi novamente ratificado o ZigBee-Pro de forma a adicionar funcionalidades de segurança melhoradas, como por exemplo a derivação de chaves, e o envio de mensagens com maior dimensão através de fragmentação. A partir deste momento passou a ser possível suportar 64000 equipamentos na mesma rede. Em 2012 houve uma nova ratificação com o nome de ZigBee-2012. Nesta versão foi introduzida uma melhor gestão de endereços e seleção de frequência automática de forma a evitar interferências. Passou ainda a ser possível usar equipamentos *Green Energy*, que se caracterizam por não ter qualquer fonte de energia, aproveitando ações de outros equipamentos, como o ligar/desligar de uma luz ou vibração para gerar energia. O ZigBee 3.0 aparece em 2016 e deixaram de existir perfis aplicacionais, embora os equipamentos ainda suportem os perfis. Perfis como o *Light Link 1.0* e o *Home Automation 1.2* estão prontos para a versão 3.0, no entanto o perfil *Smart Energy*, apesar de compatível, tem requisitos de segurança que apenas podem ser geridos pelo próprio perfil. Foi ainda criado o *ZigBee's Over-The-Air* que permite realizar atualizações nas aplicações nos equipamentos em funcionamento. Muito recentemente, em Junho de 2017, foi anunciado o ZigBee Pro 2017 caracterizando-se como a primeira rede para IoT capaz de funcionar simultaneamente em duas bandas ISM (*Industria, Scientific and Medical*).

Existem muitas dúvidas relativamente à origem do nome, mas a teoria mais consensual é que se deve às «danças» das abelhas quando estão nas colmeias.

O termo Zigbee é uma marca registada pela ZigBee Alliance, responsável por manter e publicar a norma ZigBee. Além de ser responsável pela norma, também publica perfis aplicacionais que permitem criar produtos transversais independentemente da marca.

Caso se tenha um propósito comercial, é necessário adquirir uma licença para usar a especificação ZigBee. Caso contrário, é gratuita.

O fator segurança foi tido em conta a partir do momento da conceção da norma. O uso de AES 128-bit, de diferentes modos de segurança ou diferentes chaves de cifra com diferentes propósitos são apenas algumas das medidas implementadas.

Estes padrões caracterizam-se por serem de baixo-custo, bastante simples e de extremo baixo consumo. Este último ponto é na verdade uma das maiores vantagens do ZigBee, havendo casos onde os equipamentos chegam a trabalhar cinco anos com uma única bateria [6, pp. 9].

Outra das grandes vantagens do protocolo é poder ser usado por qualquer pessoa, visto não ser propriedade de uma empresa, como acontece com o Z-Wave. Aqueles que estão interessados em usar o Z-Wave têm de pagar uma taxa à Sigma Designs, empresa proprietária deste protocolo, tornando a sua disseminação bastante mais lenta do que o ZigBee.

O ZigBee está direcionado para o mercado de casa e automação, sendo os principais focos o controlo de luz, segurança, sistemas elétricos e HVAC (Aquecimento, Ventilação e Ar-condicionado). Com a crescente adoção da tecnologia também se começou a ver a sua implementação em tecnologia *smart-grid*, especialmente em medição avançada de infraestruturas.

Os analistas apontam que o ZigBee venha a ser a tecnologia a suportar o grande crescimento das IoT dos próximos anos, esperando-se um crescimento na ordem dos 550% de uso de circuitos ZigBee até 2020 [31].

Nas secções seguintes são explicadas as várias camadas existentes no protocolo ZigBee, incluindo as funcionalidades de cada uma delas. São também detalhadas as opções de segurança implementadas e é apresentada uma comparação entre este protocolo, o Wi-Fi e o Bluetooth.

2.2 Camadas ZigBee

O ZigBee tem um total de quatro camadas, como é ilustrado na figura 1, tendo como base as camadas Física e *Media Access Control*, propostas pela IEEE 802.15.4 [51], e outras duas camadas que dão pelo nome de Rede e Aplicação.

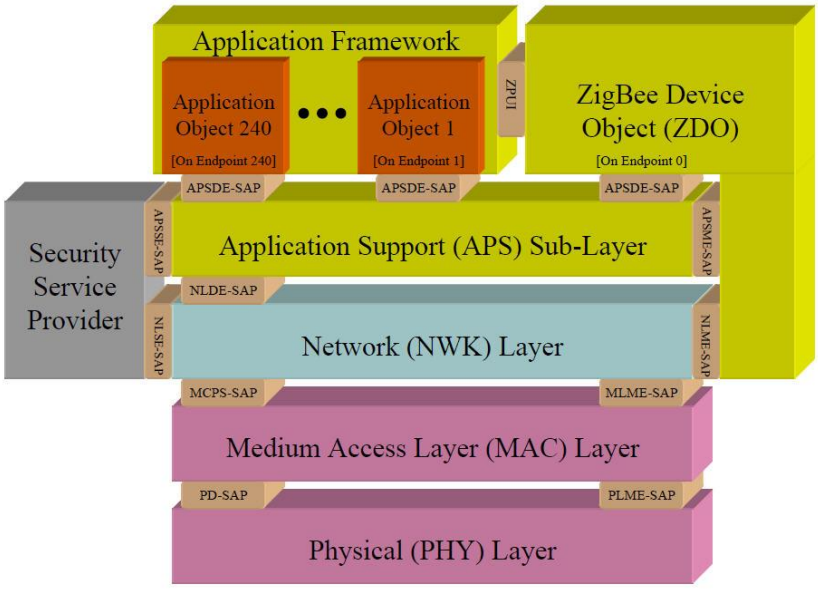

Figura 1 - Camadas ZigBee
Extraído de [30]

2.2.1 Física (PHY)

A camada física é definida na norma IEEE 802.15.4 [51]. Esta camada permite usar várias frequências diferentes dependendo do continente onde opera, existindo a frequência 2.4GHz dedicada a operações no mundo inteiro.

Apesar da existência de frequências regionais, muitos dos fabricantes preferem usar a frequência internacional pelo facto de conseguirem garantir que o mesmo produto pode ser vendido em diferentes mercados. No entanto, o uso de frequências menores garante uma maior eficácia devido a uma melhor propagação e maior alcance do sinal. A desvantagem no uso destas frequências prende-se com uma menor taxa de transmissão de dados.

Dependendo da frequência usada, existem canais diferentes disponíveis para a propagação de sinal. Na Europa apenas existe o canal zero disponível, enquanto nas frequências do continente americano existem dez canais, do canal um ao dez, como se pode verificar na tabela 1. A frequência internacional utiliza os canais 11 até ao 26.

O tráfego é propagado numa única frequência a não ser que seja configurado de outra forma [48, pp. 410].

Tabela 1 - Frequências e canais sub-1-GHz

Número do Canal	Frequência	Geografia
0	868 MHz	Europa
1	906 MHz	América
2	908 MHz	América
3	910 MHz	América
4	912 MHz	América
5	914 MHz	América
6	916 MHz	América
7	918 MHz	América
8	920 MHz	América
9	922 MHz	América
10	924 MHz	América
0	780 MHz	China
1	782 MHz	China
2	784 MHz	China
3	786 MHz	China

2.2.2 Media Access Control (MAC)

Esta camada também é definida pela norma IEEE 802.15.4 e permite construir redes ZigBee, onde são definidas as topologias, papéis dos equipamentos e associação/desassociação da rede.

No ZigBee a atribuição de tarefas aos equipamentos é crucial para o funcionamento da rede, havendo as seguintes distinções:

- *ZigBee Coordinator* (ZC) – Responsável por controlar a rede e realizar reencaminhamento de mensagens em nome de outros equipamentos. Permite ainda que os outros se juntem a ele e participem na rede.
- *ZigBee Router* (ZR) – É muito idêntico ao ZC em termos de *hardware*, no entanto, difere nas tarefas de gestão de rede implementadas por *software*. Faz reencaminhamento de mensagens e permite que os outros equipamentos se juntem a ele para participar na rede.
- *ZigBee Trust-Center* (TC) – Responsável pela autenticação de equipamentos na rede. Quando um equipamento se tentar ligar à rede, o *router* mais próximo vai notificar o TC das intenções deste novo equipamento. O TC diz ao *router* se deve permitir, ou não, a nova conexão. Este equipamento é geralmente considerado um ponto de falha único.

- *ZigBee End Device* (ZED) – É um equipamento limitado na sua participação na rede ZigBee, pois não consegue retransmitir mensagens para outros equipamentos. Não permite que nenhum equipamento se junte a ele e apenas comunica com ZCs e ZRs.

A figura 2 apresenta duas topologias distintas:

- Estrela – Todos os equipamentos, independentemente de serem ZR ou ZED ligam-se diretamente ao ZC. A grande vantagem é que são necessários apenas dois saltos para chegar ao destino. Quanto a desvantagens, toda a rede se baseia no ZC que pode sofrer um *Denial-of-Service* (DoS) involuntário caso haja uma troca excessiva de pacotes.
- Malha – Os ZED e ZR podem ligar-se ao ZC ou a outros ZR, não estando obrigados a ter uma conexão direta ao ZC. A característica desta topologia é o facto de ser autorregeneradora, significando que, no caso de um caminho falhar, um nó irá encontrar outro caminho, pois existe a noção de redundância de caminhos. Por outro lado, é um protocolo de encaminhamento mais complexo que a topologia em estrela, sendo necessário uma maior sobrecarga de pacotes para se realizar um DoS.

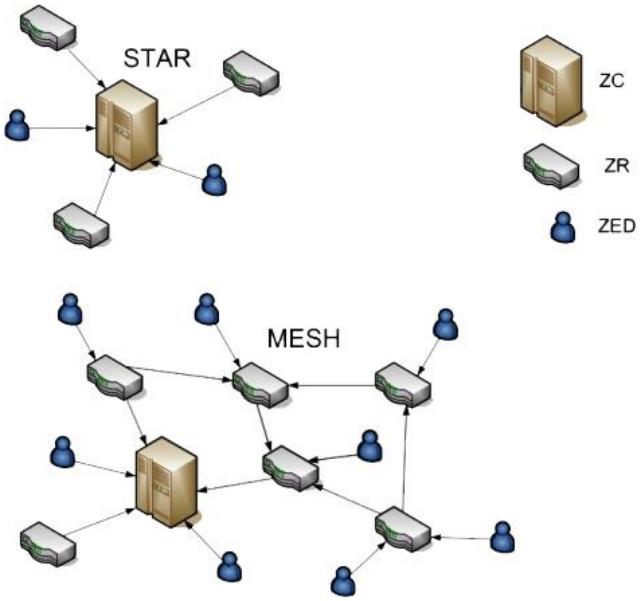

Figura 2 - Topologias ZigBee
Extraído de [54]

Em ambos os casos, o ZC tem um papel autoritário no que toca à gestão da rede.

A norma IEEE 802.15.4 prevê a existência da topologia em árvore, mais concretamente da topologia cluster-tree. No entanto, o ZigBee não suporta esta topologia.

As comunicações ZigBee são transportadas nesta camada pelos seguintes *frames*:

- *Beacon Frame* – Usadas para varrer a rede para descobrir ZRs e ZCs.
- *Data Frames* – Usadas para trocar dados entre equipamentos. Têm *payload* máximo de 114 bytes
- *Ackowledgement Frames* – Caso pedido, é enviada uma *acknowledgement frame* para indicar que uma *frame* foi recebida.
- *Command Frames* – Usadas para controlar operações de rede como associação, resolução de conflitos e pedidos de entrega de dados pendentes.

A tabela 2 apresenta o formato da *frame* MAC.

Tabela 2 - Formato da Frame MAC.
Extraído de [6]

Octets:2	1	0/2	0/2/8	0/2	0/2/8	Variable	2
Frame Control	Sequence number	Destination PAN ID	Destination address	Source PAN ID	Source address	Data payload	FCS
		Data payload					
MAC header						MAC payload	MFR

2.2.3 Rede (NWK)

A camada de rede é definida na especificação ZigBee e é responsável pela formação de rede, descoberta de equipamentos, atribuição de endereços e encaminhamento.

A tabela 3 apresenta o formato da *frame* NWK.

Tabela 3 - Formato da frame NWK
Extraído de [6]

Octets:2	2	2	1	1	Variable
Frame Control	Destination address	Source address	Radius	Sequence number	Data payload
	Routing fields				
NWK header					NWK payload

2.2.4 Aplicação (APS)

A camada superior também é definida pelo ZigBee e tem como objetivo especificar a operação e interface para objetos aplicacionais que definem as funcionalidades de um equipamento.

Os objetos aplicacionais podem ser escolhidos a partir dos perfis ZigBee previamente criados pela ZigBee Alliance. Esses perfis também podem ser desenvolvidos para uso em equipamentos proprietários. Os perfis atualmente desenvolvidos ou em finalização são os seguintes:

- *Home Automation*
- *Smart Energy*
- *Telecommunication Services*
- *Health Care*
- RF4CE – *Remote Control*
- RF4CE – *Input Device*
- *Remote Control*
- *Light Link*
- IP
- *Commercial Building Automation*
- *Gateway*
- *Green Power*
- *Retail Services*
- *Zigbee Smart Energy*
- *Smart Energy*
- *Light Link*

A camada *ZigBee Device Object* (ZDO), presente dentro da camada de aplicação, é responsável por fornecer a interface necessária aos equipamentos ZigBee, incluindo o uso de serviços de segurança, onde se inclui o uso e remoção de chaves cifradas e serviços de gestão de rede.

Existe ainda uma subcamada, com o nome de *Application Support Sublayer*, que oferece as funções necessárias aos perfis, como, por exemplo, a entrega de dados de forma confiável.

A tabela 4 apresenta o formato da *frame* APS.

Tabela 4 - Formato da frame APS.
Extraído de [6]

Octets:1	0/1	0/2	2	2	1	1	0/1/2	Variable
Frame Control	Dst EP	Group address	Cluster ID	Profile ID	Src EP	APS counter	Extended Header	Data payload
	In-Node Addressing fields							
APS header								APS payload

2.3 Segurança

A segurança foi um requisito tido em conta desde o momento da criação do protocolo e melhorado ao longo das várias ratificações.

O nível de segurança depende da salvaguarda das chaves simétricas, nos mecanismos de proteção usados e da implementação apropriada dos mecanismos criptográficos e das políticas de segurança.

A norma diz que os equipamentos não podem transmitir de forma inadvertida as chaves para outros equipamentos, a não ser que haja garantia de proteção, tal como no momento de transporte de chaves. A única exceção ocorre quando um equipamento sem chave pré-configurada se junta à rede, sendo enviada uma chave em claro, criando um momento de vulnerabilidade.

Este modelo de confiança levou a que fossem criadas as seguintes regras de segurança:

- A camada que origina a *frame* é responsável por a transmitir de forma segura. Tanto a camada de aplicação APS como a camada de rede NWK são capazes de proteger a *frame*, tendo em conta requisitos de confidencialidade e integridade.
- Se for necessária proteção de roubo de serviço, os mecanismos de segurança da camada NWK têm de ser usados em todas as *frames*, com exceção das *frames* que passam entre o *router* e um equipamento recém-chegado. Esta camada vai providenciar garantias de confidencialidade e integridade às camadas superiores.
- A segurança pode-se basear na reutilização de chaves por cada uma das camadas. Esta reutilização ajuda a reduzir custos de armazenamento.
- É garantida segurança ponto-a-ponto com base em criptografia.
- Todos os equipamentos têm de usar o mesmo nível de segurança. Caso uma aplicação necessite de um nível de segurança mais elevado, deve criar uma nova rede com um nível de segurança mais elevado.

De forma a garantir que uma implementação trate os seus problemas corretamente, os perfis aplicacionais são responsáveis por:

- Detetar e tratar a dessincronização das chaves.
- Detetar e tratar a dessincronização ou *overflow* dos contadores.
- Caso seja necessário, ter métodos de atualizações periódicos e revogação de chaves.

Existem três tipos de chaves:

- Chave de Ligação – Chaves de 128-bit, partilhadas entre dois equipamentos, usadas para comunicações *unicast* entre eles. Esta chave deve ser partilhada por método de transporte, de pré-instalação ou de estabelecimento. Existem dois tipos de chaves de ligação: global e única.
- Chave de Rede – Chaves de 128-bit, partilhadas por todos os equipamentos, usadas para comunicações de *broadcast*. Esta chave deve ser partilhada por método de transporte ou pré-instalação.
- Chave Mestra – É apenas obrigatória em ZigBee-Pro, sendo opcional nas restantes. Usada em conjunto com o *ZigBee Symmetric Key-Key Establishment* (SKKE) para derivar a chave de ligação, através de um processo de três passos. A chave mestra pode ser partilhada por método de transporte, de pré-instalação ou pela inserção de dados pelo utilizador.

A norma ZigBee [51] diz que deve ser definida uma *global trust center link key* caso nenhuma chave de ligação esteja definida pela aplicação na altura de emparelhamento. Essa chave por omissão deve ter o seguinte valor:

- 5A:69:67:42:65:65:41:6C:6C:69:61:6E:63:65:30:39 (ZigBeeAlliance09)

Nas pesquisas realizadas [52][53], foi encontrada outra chave por omissão para o perfil de Light Link:

- 9F:55:95:F1:02:57:C8:A4:69:CB:F4:2B:C9:3F:EE:31

O uso de uma chave por omissão torna-se mais uma vulnerabilidade do que uma medida de segurança, pois dá acesso à chave de rede em claro.

Existem três métodos distintos para partilha de chaves:

- Transporte – As chaves de ligação, de rede e mestra são passadas em claro por rede sem fios, no momento da aprovação do acesso à rede.
- Pré-Instalada – As chaves vêm configuradas de fábrica. É um método que se revela impraticável quando é necessário alterar a chave em momentos futuros, pois a localização dos equipamentos nem sempre é acessível, ou pela falta de conhecimento do utilizador em fazê-lo.
- Estabelecimento – Mecanismo da subcamada APS que permite que dois equipamentos derivem uma chave de ligação, a partir de um elemento de confiança (chave mestra).

A norma define dois modos de segurança que controlam a forma como a chave de rede é distribuída, podendo ainda controlar a inicialização dos *frame counters*. Esses modos são os seguintes:

- Modo de Segurança Padrão – Usado para aplicações residenciais. Um equipamento comunica com o TC de forma segura, através da chave de rede pré-configurada ou é esta enviada em claro pelo método de transporte. O TC pode manter uma lista de equipamentos, chaves mestras, de ligação e de rede de todos os equipamentos na rede, contudo, tem de manter uma chave de rede padrão e controlar a autenticação de rede. Todos os equipamentos que se ligam à rede têm de ter uma chave de ligação global ou única que seja do conhecimento do TC.
- Modo de Segurança Elevado – Usado para aplicações comerciais. Um equipamento pode ser carregado com o endereço do TC e a chave mestra inicial. Caso se possa tolerar um momento de vulnerabilidade, a chave mestra pode ser enviada em claro pelo método de transporte. Caso não tenha sido previamente carregado com o endereço do TC, o endereço será o do ZC ou um equipamento designado por ele. O TC é responsável por manter a lista de equipamentos, chaves mestras, de ligação e de rede que necessita para controlar e aplicar as políticas de atualização de rede e autenticação. O TC também é responsável por implementar o SKKE.

São ainda implementados controlos de autenticidade a nível das mensagens usando o CCM*, versão modificada do AES-CCM (*Counter Mode With Cipher Block Chaining Message Auhtentication Code*). O CCM* pode garantir confidencialidade e/ou integridade.

A nível de integridade é usado o *Message Integrity Check* (MIC) que valida os conteúdos do *frame*. É possível criar mecanismos contra ataques de força bruta usando um MIC mais longo, para casos que o atacante envie *frames* alteradas. A desvantagem deste mecanismo é o consumo de ciclos de CPU e o tamanho da *frame*.

2.4 Comparativo com outros protocolos

A comparação com o Wi-Fi e o Bluetooth é praticamente obrigatória, já que para o utilizador nem sempre é fácil perceber a necessidade de adoção de mais uma tecnologia, aparentemente igual.

A grande vantagem deste protocolo é o seu consumo diminuto diretamente associado à baixa velocidade de transferência de dados que se situa entre os 20Kbps e os 250Kbps. Esta ordem de valores torna-se importante quando comparado com o Wi-Fi ou o Bluetooth, com velocidades padrão de 54Mbps e 1-3Mbps respetivamente, que obrigam a um maior consumo. Outras vantagens são a transferência em distâncias na ordem dos 10-100 metros e o facto de ser um protocolo muito simples, combinando num único circuito integrado a *stack* ZigBee (até 120KB na NVRAM), transmissores *wireless* e microprocessadores. Outro dos grandes motivos para este baixo consumo deve-se à capacidade dos equipamentos entrarem em hibernação, desligando os transcetores por várias horas em caso de necessidade. O equipamento é capaz de se voltar a ligar autonomamente, transmitir os dados e voltar a desligar-se de seguida. Visto que os coordenadores e os *routers* precisam de estar sempre contatáveis, costumam ser implementados com corrente constante.

A relação entre os vários protocolos pode ser analisada na tabela 5, onde são comparados tipos de rede, velocidades, distâncias e autonomia de bateria.

Tabela 5 - Diferenças nos vários protocolos

	Tipo de Rede	**Velocidade**	**Distância**	**Bateria**
ZigBee	*Home Area Network (HAN)*	20-250 Kbps	10-100 metros	5 anos
Wi-Fi	*Local Area Network (LAN)*	54 Mbps	50-90 metros	8-12 horas
Bluetooth	*Personal Area Network (PAN)*	1-3 Mpbs	30-35 metros	16 dias

2.5 Vulnerabilidade e ataques

Neste subcapítulo são descritos os vários ataques e vulnerabilidades existentes na rede ZigBee.

Também aqui são abordadas as várias ferramentas existentes, com especial incidência na *framework* KillerBee [33], pois são poucas as ferramentas nesta área que permitem estudar o protocolo.

2.5.1 Descoberta de rede

Esta técnica consiste em descobrir equipamentos ativos, obtendo informação relevante para o funcionamento da rede.

Existem dois métodos distintos para realizar a descoberta. O primeiro baseia-se numa técnica similar à usada em redes Wi-Fi, onde é transmitido um *beacon* num canal e aguarda por respostas dos ZCs e ZRs. Com este método obtêm-se o PAN ID, o endereço de origem, o *stack profile*, a versão de *stack* e informação do endereço IEEE. O segundo método, que tem por base o primeiro, analisa a força do sinal do último pacote recebido, conseguindo assim identificar a localização física aproximada dos diferentes equipamentos.

As ferramentas que podem ser utilizadas para este objetivo são o zbstumbler, para o primeiro método, e o zbfind, para o segundo. Ambos os aplicativos pertencem à *framework* KillerBee.

Vários autores explicam como usar estas ferramentas. É o caso do Joshua Wright e do Johnny Cache [48, pp. 426] [48, pp. 452], Wright [46] [47], Tony Lee [18], Renaud Lifchitz [19] e Olawumi, Haataja, Asikainen, Vidgren e Toivanen [24].

Nenhum destes métodos pode ser evitado visto que o mecanismo de *beacon* não pode ser desativado.

2.5.2 Análise de pacotes

Este ataque, também conhecido na comunidade de segurança como *eavesdropping*, baseia-se na escuta das comunicações sem o consentimento das pessoas alvo do ataque.

Quando se tenta escutar comunicações na rede ZigBee, é necessário adquirir equipamento adequado que consiga receber os pacotes que passam na rede, um método similar ao modo promíscuo do Wi-Fi. Para conseguir comunicar e escutar neste protocolo, é necessário adquirir e configurar *hardware* específico (RZUSBstick ou Api-Mote). No caso do RZUSBstick, Joshua Wright e Johnny Cache explicam como realizar a instalação do *firmware* alterado, necessário para correr o KillerBee [48, pp. 417]. Já para o Api-Mote, é preciso construir a placa de raiz, tendo em conta que não é um USB *stick* comercializado por uma grande marca. A Rive Loop Security tem um manual com o procedimento [12]. Esta empresa também comercializa a placa já assemblada e com o *firmware* instalado.

Existem duas ferramentas similares com o objetivo de capturar comunicações que se diferenciam principalmente pela existência de um modo de visualização gráfico:

- Zbdump – ferramenta similar ao tcpdump, dispondo toda a informação das comunicações no terminal [48, pp. 432] [40].
- Zbwireshark – Similar ao zbdump mas permite captura através do Wireshark, usando uma consola gráfica [18].

Para este ataque não existem medidas de prevenção possíveis. Em qualquer momento, um agente malicioso pode escutar as comunicações. Tendo consciência desta limitação, deve-se usar métodos de cifra com eficácia comprovada e uso de chaves fortes, mantendo-as secretas.

2.5.3 Captura de chave criptográfica

Dependendo do nível de segurança aplicado e da forma como é passada a chave, pode ser possível escutar a chave, decifrando de seguida os conteúdos apanhados com as ferramentas de análise de pacotes.

Esta diferença, relativamente aos níveis de segurança, é discutida na secção 2.3 e em [24]. Se for usado o nível de segurança elevado, não é comum a chave mestra ser transmitida em claro. Por norma, a chave de ligação é criada a partir da chave mestra pré-instalada, sendo depois enviada a chave de rede cifrada com a chave de ligação. Só a partir desse momento os equipamentos passam a confiar nas suas comunicações. No caso de ser usado o nível de segurança padrão, a chave de ligação e de rede são passadas em claro na primeira comunicação ou são definidas no momento de fabrico. Caso se use uma chave de ligação global, existem chaves por omissão, responsáveis por cifrar a chave de rede, criando um momento de vulnerabilidade.

A ferramenta mais indicada para este ataque é o zbdsniff que recebe como argumento vários ficheiros capturados e encontra a chave e os endereços MAC de destino e origem automaticamente. Joshua Wright e Johnny Cache [48, pp. 439], Tony Lee [17] e vários outros autores [42] explicam este ataque.

Também pode ser usado o zbwireshark [18], no entanto essa ferramenta obriga a uma pesquisa manual dos conteúdos.

Caso a chave venha pré-instalada de «fábrica», o método de captura obriga a um *dump* do *firmware*. Para fazer a extração do *firmware* tem de se usar a placa GoodFET, criada por Travis Goodspeed. [7]

A GoodFET funciona no protocolo JTAG que permite interagir com *chips* de modo a fazer *debug* dos equipamentos. Aproveitando esta funcionalidade, é possível extrair conteúdos da RAM para um ficheiro, visto que não é possível proteger a memória.

Estes testes foram efetuados nos *chips* da Texas Instruments CC2430, contudo Travis Goodspeed diz em [7] que os circuitos da mesma família (CC1110, CC2431, CC2510, CC2511, CC2530 e CC2531), apesar de não terem sido testados, devem estar vulneráveis.

2.5.4 Ataque de repetição

Este ataque consiste em repetir mensagens já escutadas.

Para realizar este método com sucesso é necessário estruturar o ficheiro capturado com cuidado, retirando todas as *frames* desnecessárias, para que este não falhe.

A ferramenta usada é o zbreplay que permite ler um ficheiro capturado anteriormente e retransmitir todas as *frames* com um atraso definido pelo utilizador.

Vários autores descrevem este método que pode ser consultado em [48, pp. 436], [18], [46] e [24].

Existe uma limitação a este ataque. Será necessário que não exista um número de sequência nas *frames*.

2.5.5 Ataque de disponibilidade

Um ataque de disponibilidade consiste em comprometer a rede ou qualquer um dos equipamentos de forma a que não haja uma resposta adequada aos pedidos do utilizador.

Existem várias formas de fazer este ataque, contudo o mais comum é o envio de uma quantidade exagerada de pedidos à rede fazendo com que o *router* deixe de responder.

Dentro da *framework* KillerBee a aplicação recomendada é a zbassocflood, podendo ser visto o seu uso, embora sem sucesso, por Tony Lee em [18].

Outro ataque possível, é o aproveitamento de uma falha na gestão do *frame counter* (FC) dos pacotes de IEEE 802.15.4. Esta vulnerabilidade foi descoberta por dois portugueses, Silva e Nunes [36], que se aperceberam que os nós que recebiam um pacote apenas verificavam se o FC era maior que o anterior. Tendo em conta que o FC tem um valor máximo de 0xffffffff-1, ao ser enviado esse valor, o equipamento ficava em modo *blacklist*. Para ser removido desse modo, o administrador tem de mudar a chave de rede em todos os equipamentos.

Os ataques de ruído também são uma forma de ataque de disponibilidade, contudo, devido à sua especificidade são descritos na secção 2.5.7.

2.5.6 Falsificação de pacotes

A falsificação de pacotes baseia-se na alteração de dados existentes nos pacotes transmitidos na rede.

A título de exemplo, quando se envia um pedido para a *plug* AC ligar um aparelho de ar-condicionado é enviado um código de infravermelho para a tomada. Caso se consiga alterar esse código no pacote que é enviado, passa a ser possível ligar/desligar qualquer outro equipamento que funcione com códigos infravermelhos.

Podem-se usar métodos manuais de injeção de pacotes na rede, ou então, através da *framework* KillerBee, onde é possível usar um aplicativo que permite falsificar pacotes – zbscappy. Este aplicativo é a versão do Scappy para ZigBee. O zbscappy permite escutar e alterar os pacotes que passam na rede usando o Api-Mote.

Além da documentação do KillerBee [33], os únicos autores que mencionam e descrevem com algum grau de detalhe este ataque são Joshua Wright e Johnny Cache [48, pp. 441].

A forma de mitigar este ataque prende-se com a aposta em métodos que mantêm a integridade dos pacotes.

2.5.7 Ataque de ruído

Um ataque de ruído é a inserção propositada de ruído digital na rede, afetando parcial ou totalmente o envio de comunicações.

Além do equipamento especializado para anular sinais da mesma frequência [15], também é sabido da existência de vários problemas de ruído na frequência 2.4GHz [3], como a sobreposição realizada por micro-ondas, telefones sem fios, *routers* Wi-Fi e outros equipamentos.

Bin Yan diz que o ZigBee pode ser afetado nas primeiras versões por este problema, principalmente com telefones sem fios [49].

Contudo, nas versões mais recentes, o ZR, ao criar a primeira ligação, faz uma verificação do canal mais adequado para a rede. Após o estabelecimento da rede, o protocolo usa uma função de agilidade para verificar o aumento de ruído na rede e pode recalcular um novo canal [20].

Apesar disso, Joel Crane afirma que existem interferências caso a escolha de canais Wi-Fi e ZigBee não sejam cuidadas [5].

2.6 Ferramentas para deteção de vulnerabilidades

No caso do ZigBee existem três pessoas a ter em conta, duas delas consideradas cruciais para o desenvolvimento de ferramentas de segurança para este protocolo – Travis Goodspeed e Joshua Wright – e um terceiro elemento que tem sido fundamental para a atualização das ferramentas criadas por Joshua Wright, que dá pelo nome de Ryan Speers.

Joshua Wright é o criador da ferramenta KillerBee [33], onde através da modificação do *firmware* das placas RZUSBstick da Atmel [48, pp. 417] consegue fazer auditorias de segurança ao protocolo ZigBee. O KillerBee tornou-se a base dos testes de segurança, sendo muitas vezes aproveitado por outras pessoas para tentarem melhorar ou criar novas funcionalidades. Mais tarde, Wright escreveu parte de um dos livros mais importantes de ataques a redes sem fios [48], onde detalha como instalar o *firmware* na placa da Atmel e realizar vários ataques, testes de segurança e ainda fornece medidas de mitigação. Joshua Wright não tem artigos publicados, no entanto podemos encontrar uma apresentação sua com alguns vídeos online [46][47].

Nos últimos anos, Wright deixou de trabalhar no KillerBee e este passou a ser suportado por Ryan Speers e pela sua empresa River Loop Sec. Ryan Speers é um jovem que se especializou na área de ZigBee, criando uma placa *open-source* que dá pelo nome de Api-Mote [32]. Esta placa garante uma maior estabilidade de uso, com a vantagem de vir já com o *firmware* correto instalado, não necessitando de uma programadora JTAG, como no caso da RZUSBstick. A disponibilização de um dispositivo, em jeito de «faça-você-mesmo», é uma grande vantagem, pois permite que se aumente o uso de ferramentas de segurança nesta área. No entanto, um dos circuitos está proibido de ser exportado dos Estados Unidos da América, limitando assim o seu uso.

Com o mesmo tipo de funcionalidades do RZUSBstick e do API-Mote são os Tmote Sky/TelosB. Estes equipamentos estão descontinuados e são difíceis de encontrar, levando a que sejam pouco usados e havendo pouca informação atual disponível.

Outro equipamento de referência é o criado por Travis Goodspeed que serve para extrair chaves criptográficas da memória dos *chips* com tecnologia ZigBee. O autor explica como extrair chaves diretamente da memória dos equipamentos através do seu equipamento conhecido por GoodFET [7][8].

Tobias Zillner apresentou na BlackHat, em agosto de 2015, uma nova ferramenta dedicada ao ZigBee [55]. Esta ferramenta dá pelo nome de SecBee e parece ser uma das ferramentas mais estáveis na atualidade, no entanto peca pela necessidade de usar um

Universal Software Radio Peripheral (USRP), elevando o preço da solução para mais de 1000€. O SecBee assenta totalmente no KillerBee.

Existem outros equipamentos que podem ser tidos em conta:

- Programadores JTAG – Equipamento para programar ou fazer *debug* de sistemas embebidos.
- *Microchip ZENA Network Analyzer* – Equipamento para capturar tráfego IEEE 802.15.4.
- *Sewio Networks Open Sniffer – Sniffer* para gamas abaixo do 1GHz

A nível de sistema operativo para testes de intrusão, o Kali é a distribuição de Linux mais conhecida no mercado para este ramo, sendo a escolha óbvia para um laboratório. Muitas das ferramentas usadas já vêm instaladas nesta distribuição o que facilita o início dos testes.

O KillerBee é composto pelos vários scripts em python:

- Zbid – Identifica as interfaces disponíveis que podem usar o KillerBee.
- Zbdump – Ferramenta idêntica ao tcpdump mas para captura de *frames* IEEE 802.15.4.
- Zbwireshark – Similar ao zbdump, permite captura em tempo real no Wireshark.
- Zbreplay – Implementa um ataque de repetição através da retransmissão de um ficheiro com a captura de pacotes.
- zbstumbler – Ferramenta para descoberta de equipamentos que funcionam com o protocolo IEEE 802.15.4.
- zborphannotify – Faz *spoofing* da notificação de um pacote órfão de forma a testar o comportamento do PAN *Coordinator*.
- zbrealign – Faz *spoofing* de uma *frame* de realinhamento do coordenador a um equipamento alvo.
- Zbfakebeacon – Faz *spoofing* das *beacon frames*, através de spam ou em resposta a um *beacon request*.
- zbassocflood – Faz associações repetidas para o PANID alvo de forma a causar um crash no equipamento devido a muitas conexões ao mesmo tempo.
- zbdsniff – Captura tráfego zigbee e procura pelas *frames* NWK e fornecimento de chaves.
- zbgoodfind – Faz procura de chaves em pacotes de captura cifrados. Ferramenta usada também com o GoodFET.
- zbwardrive – Descobre automaticamente interfaces nos vários canais existentes.
- zbscapy – Fornece uma *shell* interativa de Scapy.

Outra ferramenta essencial num laboratório de segurança é o Wireshark. É uma ferramenta de captura de pacotes que permite ao utilizador observar a informação que está a ser transmitida pelos equipamentos estudados.

Para o caso específico do ZigBee, o Wireshark tem de ser lançado pelo KillerBee, através do zbwireshark, pois esta aplicação não vem com a capacidade de interpretar o protocolo ZigBee de raiz.

2.7 Conclusão

Neste capítulo foi abordado o ZigBee enquanto protocolo. Foram detalhadas as quatro camadas existentes e os métodos de segurança implementados nas várias camadas.

Por uma necessidade de compreensão das razões de existência de um protocolo como o ZigBee, foi realizada uma comparação entre tecnologias idênticas, como é o caso do Wi-Fi e do Bluetooth.

Foram enumeradas as vulnerabilidades possíveis de se explorar, como é o caso de captura de chaves criptográficas ou ataques de repetição, e quais as ferramentas disponíveis para testar o protocolo.

No capítulo seguinte iremos ver o caso de estudo da plataforma re:dy, dando ênfase às vulnerabilidades ZigBee anteriormente descritas, às metodologias aplicadas e às ferramentas usadas.

3 Caso de estudo - Estudo das Vulnerabilidades da Plataforma re:dy

Este capítulo apresenta o caso de estudo das vulnerabilidades do protocolo ZigBee, no qual foram utilizados equipamentos da plataforma re:dy da EDP.

3.1 Plataforma re:dy

Re:dy vem de *Remote Energy Dynamics* e tem como principal objetivo controlar de forma remota a rede elétrica de casa do seu utilizador, ao mesmo tempo que permite analisar os consumos dos equipamentos elétricos que estejam conectados a esta plataforma.

A importância deste sistema da EDP tem vindo a ser destacada através das várias nomeações e prémios que tem recebido, nomeadamente o *Utility Initiative Award Nominee* em 2015 e *World Summit Winner* 2016.

Das várias funcionalidades existentes neste sistema, destaca-se a capacidade de obter informação sobre os equipamentos ligados ao re:dy, através do *smartphone* ou computador; controlar remotamente equipamentos, permitindo ligá-los ou desligá-los; receber alertas em casos de anomalia ou consumos inesperados; criar modos de utilização e agendamento dos equipamentos; ter noção do consumo na totalidade e por equipamento e simular a presença em casa através de agendamentos automáticos, sendo esta uma grande vantagem a nível da segurança física e pessoal.

Com as funcionalidades já descritas, o re:dy permite controlar o sistema elétrico ao mesmo tempo que aumenta o grau de segurança que o utilizador tem sobre a sua casa. Este sistema tem sido uma aposta da EDP para produtores de energia através do sistema solar que, desta forma, conseguem um maior controlo da sua produção. Não deixa de ser interessante controlar equipamentos elétricos à distância, assim como conseguir ligar/desligar caldeiras e bombas de calor. Também é possível desligar equipamentos de ar-condicionado através de infravermelhos. Uma clara aposta no futuro é a possibilidade de analisar os consumos associados ao carregamento de um carro elétrico.

3.1.1 Arquitetura re:dy

A arquitetura re:dy consiste na coexistência de várias tecnologias distintas, havendo um destaque para o ZigBee.

Em casa do cliente existe um conjunto de equipamentos ZigBee que se ligam ao encaminhador ZigBee (designado por re:dy *box*) através de uma topologia em malha, permitindo que existam caminhos redundantes. O único equipamento que é exceção nesta rede é o Meter que funciona em PLC. O re:dy box permite a ligação à Internet e, desta forma, ao *website* que se encontra nos servidores da EDP.

Em caso de falha da ligação à Internet, a informação de consumo é mantida para envio posterior.

A figura 3 ilustra a arquitetura da plataforma re:dy.

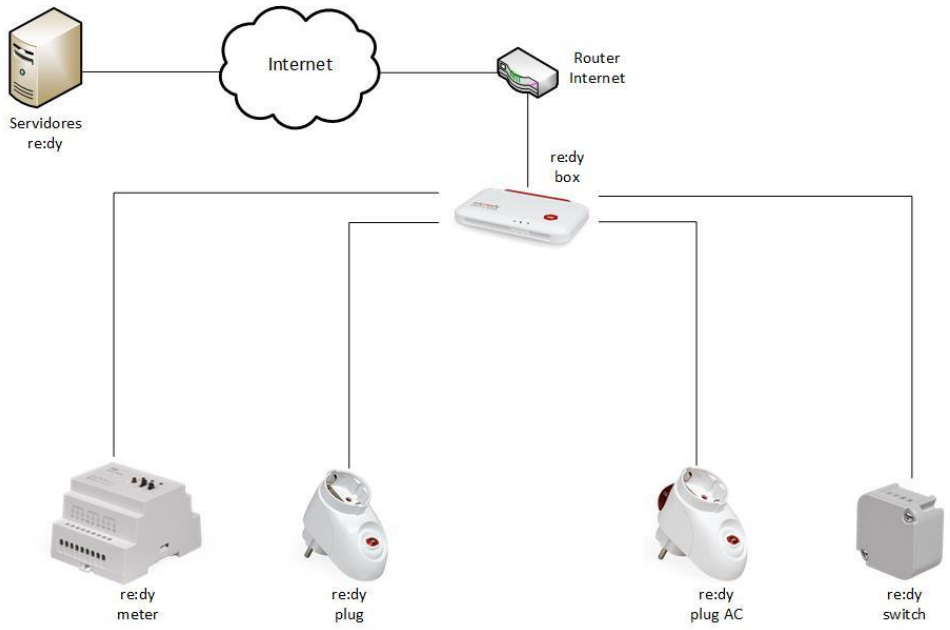

Figura 3 – Arquitetura da plataforma re:dy

3.1.2 Equipamentos re:dy

O re:dy é composto pelos seguintes equipamentos:

a) re:dy Box

Para fazer a ligação à rede IP, a EDP dispõe da re:dy box, na figura 4, que não é mais que um *ZigBee Coordinator*, que também funcionará como *ZigBee Trust-Center*. Assemelha-se fisicamente a um equipamento *router* de Internet. Este equipamento é responsável por coordenar a rede ZigBee, como descrito na secção 2.2.2.

Figura 4 - EDP re:dy ZigBee Coordinator

b) re:dy Plug

Existem diferentes *re:dy plugs*, havendo poucas diferenças entre elas.

A principal é a *plug* simples, presente na figura 5, que é uma tomada inteligente que permite ligar qualquer equipamento elétrico ao sistema re:dy. Basta colocar a *plug* numa tomada normal e ligar o equipamento para passar a respetiva informação para o sistema.

Passa a ser possível controlar e programar o seu funcionamento remotamente e conhecer o consumo individual de cada equipamento.

Figura 5 - re:dy plug simples

As *plugs* A/C, na figura 6, têm a particularidade de controlar os aparelhos de ar-condicionado remotamente por infravermelhos, a partir de uma lista pré-configurada pela EDP.

Figura 6 - re:dy plug A/C

Existem ainda as *plugs* solares que são especialmente desenhadas para sistemas de energia solar, permitindo monitorizar a produção de energia, sabendo quanto dinheiro se está a ganhar, avisando também caso haja algum problema na produção.

A aparência exterior é exatamente idêntica ao *plug* normal.

Em todas as *plugs* existe um botão que serve para ligar/desligar da corrente elétrica ou para funções mais avançadas de emparelhamento com a re:dy box. Quando se encontram desligadas da corrente, a luz do botão é um pouco mais fraca.

Para garantir que não existe perda de informação e que o sistema consegue estar 100% conectado, as *plugs* comportam-se como *ZigBee Routers*, permitindo receber novos nós como «filhos», encaminhando a informação para estes quando necessário.

c) re:dy Switch

A sua principal função é o controlo de caldeiras ou bombas de calor, embora com algum conhecimento em eletricidade facilmente se possa adaptar qualquer equipamento elétrico.

O funcionamento para o utilizador é idêntico às *plugs*, no entanto a aparência e o botão de funções é diferente.

Pode-se ver o design na figura 7 e tal como as restantes *plugs*, este também é um *ZigBee Router*.

Figura 7 - re:dy Switch

d) re:dy Meter

O re:dy meter, na figura 8, tem como objetivo o controlo energético de equipamentos encastrados, circuitos de iluminação, bombas de água ou sistemas de rega. De todos os equipamentos existentes no re:dy, este é o único que se carateriza por ser um *Programmabe Logic Controller* (PLC).

Figura 8 - re:dy Meter

3.1.3 Aplicação *web*

A aplicação web tem duas interfaces: o *website* e a aplicação móvel.

No *website*, mostrado na figura 9, o utilizador tem acesso completo a toda a informação disponibilizada pelo sistema re:dy. Neste, podem ser consultados detalhes sobre os consumos diários e mensais, impacto ambiental, potência real consumida, como também interagir com os equipamentos, definir simulações de presença e alertas.

Na aplicação móvel, presente na figura 10, apesar de ter um aspeto totalmente diferente, em pouco difere do *website*.

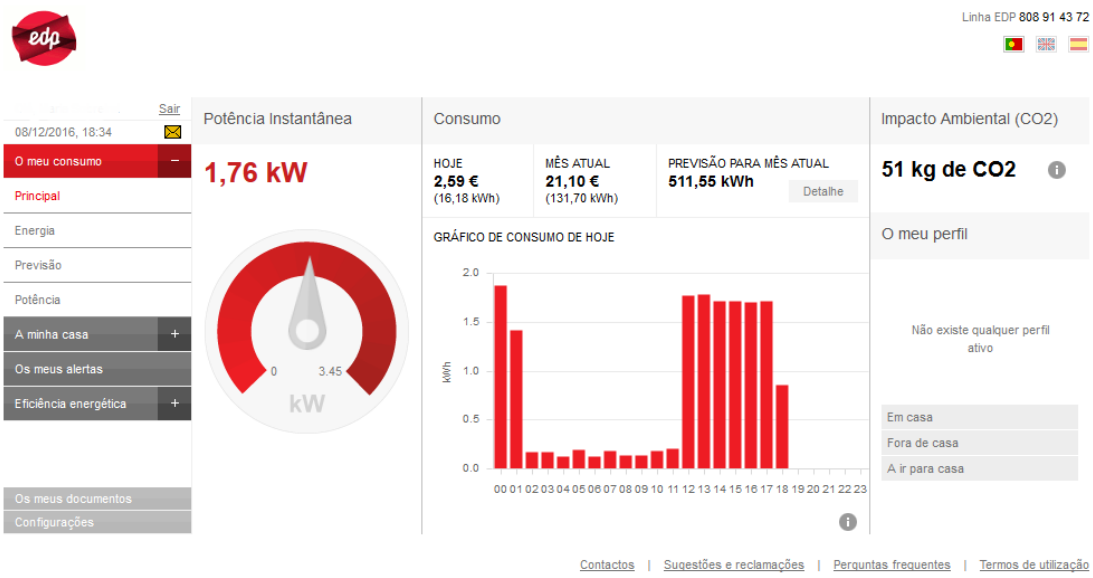

Figura 9 - Interface gráfico Web

Figura 10 - Interface gráfico mobile

3.2 Laboratório

Esta secção descreve a rede ZigBee assim como o ambiente de laboratório (*hardware* e *software*) utilizados para determinar as vulnerabilidades da rede ZigBee da plataforma re:dy.

3.2.1 Rede e equipamentos

A rede ZigBee testada inclui os seguintes dispositivos:

- 1x re:dy box
- 2x re:dy plug
- 1x re:dy plug A/C
- 1x re:dy switch

3.2.2 Laboratório de testes - *hardware*

O laboratório de testes é constituído pelos seguintes equipamentos:

a) Portátil HP EliteBook

Devido à necessidade de correr testes demorados e contínuos, foi conveniente ter um equipamento em permanente ligação com a plataforma re:dy.

Esta máquina foi responsável pela maioria dos testes no protocolo ZigBee e simulação de ataques de força bruta.

b) Portátil Toshiba Satellite P850-31N

O segundo portátil servia para um maior poder de computação em certos testes e para a deslocação à EDP aquando a realização dos testes de intrusão nas páginas web.

c) Monitor Asus VW195S

Um monitor de 17 polegadas aliado ao HP EliteBook permitiu uma análise mais rápida da captura de pacotes no Wireshark, além de toda a agilidade que permite em termos de organização de trabalho.

d) RZUSBstick

Numa primeira fase foram usadas duas *pens* RZUSBstick. Estas *pens* permitem comunicar no protocolo ZigBee.

Para que seja possível realizar testes de intrusão com o KillerBee, as RZUSBstick precisam de ser reinstaladas com um *firmware* alterado. Apesar da instalação com sucesso do *firmware*, as pens nunca tiveram o comportamento desejado tendo sido necessário a adquirir um Api-mote.

Para realizar a instalação do *firmware* usei a programadora JTAG Atmel-ICE. Como esta programadora nunca tinha sido usada para este propósito, foi-me pedido por Ryan Speers que detalhasse o processo de instalação, para publicitar no GitHub oficial do KillerBee. Com base na informação existente para a programadora Atmel AVR Dragon [33] criei o método de instalação para a Atmel-ICE, possível de ser consultado no Anexo I.

Devido aos erros encontrados e reportados no decorrer da realização deste trabalho, foi lançada uma nova versão de *firmware.* Essa nova versão foi testada e reportada como estável aos criadores da mesma.

e) Api-Mote

O Api-Mote permite comunicar no protocolo ZigBee e usar o *framework* KillerBee de forma a realizar ataques e testes de intrusão ao re:dy.

f) GoodFET

O GoodFET é uma placa capaz de comunicar e recolher o *firmware* diretamente dos *chips* das *plugs/switches*.

Esta captura permite aceder ao sistema de ficheiros, tendo como objetivo principal descobrir a chave de cifra usada nas comunicações ZigBee.

3.2.3 Laboratório de testes - *software*

O *software* usado foi o sistema operativo Kali Linux, sendo uma das distribuições de testes de intrusão mais completas, e a *framework* KillerBee, por ser o conjunto de ferramentas existente para estudar o protocolo ZigBee. O Wireshark, e a sua versão para ZigBee, zbwireshark, são também uma escolha comum na comunidade de segurança para análise de comunicações.

Apesar do KillerBee ser uma das aplicações instaladas de raiz no Kali, foi necessário reinstalar a aplicação, visto a versão existente no GitHub ser mais recente.

3.3 Testes ZigBee

De forma a determinar as vulnerabilidades da rede ZigBee na plataforma re:dy, foi seguida a metodologia das cinco fases de um ciberataque, possível de visualizar na figura 11.

Esta metodologia carateriza-se por ter uma fase inicial de reconhecimento do alvo a atacar, seguida de uma segunda fase de pesquisa de serviços abertos, documentos expostos, versões de *software*, entre muitas outras opções, com o objetivo de procurar vulnerabilidades. A terceira fase resume-se em comprometer o sistema, que é imediatamente seguida por duas fases pós-ataque que se focam em manter uma porta aberta para futuros ataques e apagar vestígios do ataque.

Estas fases tornam-se lógicas com a experiência, contudo, não significa que tenham de ser seguidas de forma linear ou que exista a obrigatoriedade de cumprir todos os passos do ciclo para garantir o sucesso de um ataque.

Figura 11 - Fases de um ciberataque

3.3.1 Reconhecimento

A primeira fase de qualquer ataque é a recolha de informação. Nesta fase foi pesquisada informação física e lógica (modelos de placas e *chips*, endereços físicos das *plugs* e *router*, canal e *stack profile*).

a) Informação Física

Por recolha de informação física entende-se a verificação da existência de números de série, marcas, modelos dos *chips* e números de códigos de barras presentes nos equipamentos.

[Texto removido por motivos de confidencialidade].

b) Informação Lógica

Por informação lógica entende-se informação relativa às comunicações de rede, particularmente os protocolos IEEE 802.15.4 e ZigBee.

Usei as placas RZUSBstick e Api-Mote com a *framework* KillerBee, nomeadamente os *scripts* zbid e zbstumbler.

O primeiro passo é verificar se a placa está a ser reconhecida com o *firmware* correto, através do comando zbid, como pode ser visto na figura 12.

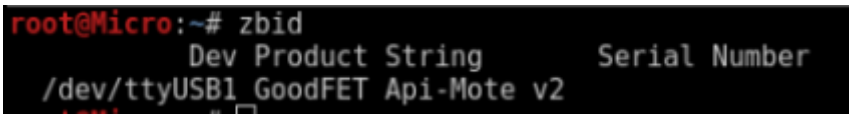

Figura 12 - Output do comando zbid

[Texto removido por motivos de confidencialidade].

3.3.2 Scanning

As ferramentas de scanning permitem obter os serviços disponíveis na rede que se está a estudar. No caso da rede ZigBee, os serviços disponíveis fazem correspondência aos perfis aplicacionais.

Tendo em conta que nenhuma das ferramentas permite saber especificamente o perfil aplicacional usado, esta etapa não se aplica.

3.3.3 Ganhar Acesso (Ataque)

Nesta secção são descritos os vários ataques ao protocolo ZigBee, sendo explicado o método de ataque, as ferramentas usadas e os resultados encontrados.

a) Análise de pacotes

[Texto removido por motivos de confidencialidade].

b) Captura de chave criptográfica

[Texto removido por motivos de confidencialidade].

c) Ataque de repetição

[Texto removido por motivos de confidencialidade].

d) Ataque de Disponibilidade

[Texto removido por motivos de confidencialidade].

e) Falsificação de pacotes

[Texto removido por motivos de confidencialidade].

f) Ataque de Ruído

[Texto removido por motivos de confidencialidade].

3.3.4 Manter acesso

O facto deste protocolo ser sem fios e não precisar de qualquer *password* ou autorização para aceder aos conteúdos que passam no «ar», faz com que seja fácil a um atacante aceder à informação que passa na rede do re:dy.

A partir do momento que se obtém a chave de rede, o acesso à informação torna-se permanente, não sendo necessário criar formas de manter o acesso.

3.3.5 Encobrir evidências

[Texto removido por motivos de confidencialidade].

3.4 Conclusão

Neste capítulo foi abordado o estudo prático das vulnerabilidades da plataforma re:dy a nível do protocolo ZigBee, tendo por base o método de cinco fases de um ciberataque.

Foi apresentada a composição dos sistema re:dy, o laboratório montado nos diferentes tipos de testes e feita a descrição de vulnerabilidades e ataques.

[Texto removido por motivos de confidencialidade].

Na realização deste trabalho, uma das grandes dificuldades sentidas prendeu-se com a falta de estabilidade das ferramentas de deteção de vulnerabilidades para ZigBee, muito devido ao funcionamento anómalo das ferramentas e da falta de documentação existente. Por este motivo tornou-se necessário criar um procedimento para instalação do *firmware*

KillerBee na RZUSBstick através da programadora JTAG Atmel-ICE, e um novo firmware KillerBee para os erros observados no realizar de alguns ataques.

No próximo capítulo será realizado o estudo na rede IP, imediatamente adjacente à rede ZigBee, e ao *website*.

4 Testes da rede IP e *website*

Como já dito anteriormente, a rede da plataforma re:dy está ligada a uma rede IP através de uma ligação *ethernet* entre o re:dy box e o *router* de Internet. Apesar deste canal não ser da responsabilidade da EDP, a informação que flui nele tem de garantir confidencialidade, de forma a que as informações de um dado utilizador re:dy não sejam observadas por um atacante. Além disso, a re:dy box terá obrigatoriamente serviços disponíveis para o exterior, aumentando assim os possíveis vetores de ataque.

Também faz parte o *website*, usado pelo cliente re:dy, para ligar/desligar as plugs e consultar toda a sua informação de consumo. Tal como na rede IP, o website tem de garantir a confidencialidade dos dados do utilizador.

4.1 Laboratório

O laboratório usado nestes testes é idêntico ao proposto nos ataques à rede ZigBee, excluindo as ferramentas destinadas unicamente a esse protocolo e com a inclusão das ferramentas descritas nas secções 4.2.2 e 4.2.3.

4.1.1 Rede e equipamentos

A rede IP testada inclui os seguintes equipamentos:

- 1x *router* Internet
- 1x re:dy box

4.1.2 Laboratório testes – *Hardware*

A nível de *hardware* é usado o Raspberry Pi 3, para fazer de Transparent Network Tap (NTAP) [2], sendo colocado entre a re:dy box e o *router* com ligação à Internet. Desta forma, é possível escutar todas as comunicações realizadas pelo sistema da EDP e compreender o seu comportamento. Tendo em conta que o Raspberry apenas tem uma porta de rede, é necessário também usar um conversor USB para *ethernet*.

4.1.3 Laboratório testes – *Software*

Além do *software* já utilizado, como por exemplo o Wireshark, é usado o sistema operativo Raspbian, com as configurações de *bridge* necessárias para realizar o ataque *man-in-the-middle.*

Para os ataques SSH descritos na seção 4.1.2 são usados o módulo SSH_Enumuser do Metasploit, para pesquisar eventuais utilizadores no OpenSSH, e o Hydra para realizar um teste de força bruta ao utilizador root.

4.2 Testes rede IP

Será aplicado a mesma metodologia dos cinco passos, presente na figura 11, usada nos ataques ZigBee.

4.2.1 Reconhecimento

Nesta fase pretende-se descobrir serviços de rede a correr na re:dy box, de forma a obter informação relevante que possa comprometer o acesso ao equipamento ou aos dados transmitidos por este.

Usando a ferramenta nmap, presente no Kali e com o resultado presente na figura 23, é possível verificar quais os IPs e descrições dos equipamentos na rede IP, o sistema operativo do equipamento testado, o endereço MAC e os portos abertos.

[Texto removido por motivos de confidencialidade].

4.2.2 Scanning

[Texto removido por motivos de confidencialidade].

4.2.3 Ganhar Acesso (Ataque)

a) Captura e Análise

[Texto removido por motivos de confidencialidade].

b) Ataques SSH

[Texto removido por motivos de confidencialidade].

4.2.4 Manter acesso

[Texto removido por motivos de confidencialidade].

4.2.5 Encobrir evidências

[Texto removido por motivos de confidencialidade].

4.3 Testes ao website

Da plataforma re:dy faz parte o website, responsável por dar ordens às plugs e por servir de meio de consulta das informações de consumo do utilizador.

[Texto removido por motivos de confidencialidade].

São simulados vários ataques contra o website de pré-produção, contudo quando não se trata de testes intrusivos devem-se realizar testes e comparar com o *website* de produção.

É de notar que as plataformas re:dy são testadas mensalmente por uma empresa de testes de intrusão, dificultando assim a descoberta de eventuais vulnerabilidades.

4.3.1 Laboratório

O laboratório usado para estes testes foi o mesmo usado nos testes descritos anteriormente, detalhado na secção 3.2. No entanto, foi usado *software* específico para este tipo de testes:

- Firefox – *Browser* que permite a visualização dos conteúdos das aplicações web.
- Burp – *Proxy* que permite a alteração de conteúdos durante as comunicações.
- Nikto – Ferramenta que verifica *headers* e más configurações nas páginas web definidas.
- DirB – Aplicação que aplica um dicionário de forma a verificar pastas e ficheiros existentes na página alvo definida.
- Google Dorks – Permitem uma pesquisa filtrada no Google de eventuais vulnerabilidades ou informações que não deveriam ser públicas.
- Whois – Ferramenta que devolve informação de um dado IP ou *hostname*.
- Shodan – Motor de busca de equipamentos ligados à Internet
- Web Scanners – Vários web scanners, como o Immuny Web [45], Asafa [12] e NetCraft [21], que permitem verificar de forma automatizada *headers* mal formatados e vulnerabilidades já conhecidas.
- Netdiscover – Para descobrir quais os IPs existentes na rede onde o *router* ZigBee está ligado.
- Nmap – Para realizar uma pesquisa mais ativa, devolvendo versões de sistema operativo, portos abertos com descrição e endereço MAC.
- Wappalyzer – *Add-on* do Firefox que devolve as tecnologias usadas no site web.
- Nessus – *Scanner* automático de vulnerabilidades web e físicas

4.3.2 Testes aplicacionais

a) Reconhecimento

[Texto removido por motivos de confidencialidade].

b) Scanning

[Texto removido por motivos de confidencialidade].

c) Ganhar acesso (Ataques)

[Texto removido por motivos de confidencialidade].

d) Manter acesso

[Texto removido por motivos de confidencialidade].

e) Encobrir evidências

[Texto removido por motivos de confidencialidade].

4.4 Conclusão

Neste capítulo foi abordado o estudo prático das vulnerabilidades da plataforma re:dy da rede IP e *website* de apoio ao sistema, tendo por base o método de cinco fases de um ciberataque.

[Texto removido por motivos de confidencialidade].

5 Conclusão

O trabalho realizado visou analisar, estudar e comprometer o protocolo ZigBee, já que é considerado uma das tecnologias de maior implementação nas *Internet of Things* (IoT), com um crescimento esperado de 550% para os próximos anos [31]. Tendo em conta que a segurança foi tida como um fator importante desde a criação da norma, as dificuldades poderiam ser à partida acrescidas, pela existência de métodos mais seguros, quando comparados com outras tecnologias similares.

Como caso de estudo, foi usado o sistema de controlo de tomadas e registo de consumos da EDP – o sistema re:dy.

Após ter sido realizado um levantamento das vulnerabilidades do protocolo e das ferramentas de deteção dessas vulnerabilidades, foi efetuada a análise ao sistema re:dy.

Um dos desafios do ZigBee é a gestão de chaves, devido às limitações computacionais dos equipamentos e da necessidade de gestão do consumo da bateria, o que obriga ao uso de chaves simétricas.

[Texto removido por motivos de confidencialidade].

Adicionalmente, a plataforma re:dy assenta numa ligação à Internet e é coordenada por uma aplicação *web*, o que significa que existiam à partida mais vetores de ataque possíveis de ser explorados.

[Texto removido por motivos de confidencialidade].

A realização deste trabalho foi dificultada pelo estado de maturidade das ferramentas, as quais exibem comportamentos inesperados, resultados incoerentes e falta de documentação. Neste contexto, destacam-se duas contribuições adicionais decorrentes deste trabalho: a descrição do processo de instalação do *firmware* KillerBee no RZUSBstick usando a programadora JTAG Atmel-ICE, pedido por Ryan Speers, e os testes ao novo *firmware*, após ter sido reportado nos fóruns apropriados os erros observados no momento da captura de pacotes.

[Texto removido por motivos de confidencialidade].

Este trabalho foi muito importante para aprofundar o conhecimento na área das IoT, no mercado de automação e no protocolo ZigBee. Permitiu ainda aperfeiçoar competências de investigação, seleção, organização, esquematização e comunicação.

6 Bibliografia

[1] A True System-on-Chip Solution for 2.4-GHz IEEE 802.15.4 and ZigBee Applications. (n.d.). Retrieved October 28, 2016, from http://www.ti.com/lit/ds/symlink/cc2530.pdf

[2] Bothered, B. (2014, May 21). Botherder/ntap. Retrieved May 20, 2017, from https://github.com/botherder/ntap

[3] Common Causes of WiFi Interference . (n.d.). Retrieved June 15, 2017, from http://packetworks.net/blog/common-causes-of-wifi-interference

[4] Cobb, S. (2017, April 25). 10 things to know about the October 21 IoT DDoS attacks. Retrieved July 27, 2017, from https://www.welivesecurity.com/2016/10/24/10-things-know-october-21-iot-ddos-attacks/

[5] Crane, J. (n.d.). ZigBee and WiFi Coexistence. Retrieved June 15, 2017, from https://support.metageek.com/hc/en-us/articles/203845040-ZigBee-and-Wi-Fi-Coexistence

[6] Gislason, D. (2008). Zigbee wireless networking. Oxford: Newnes. Retrieved 1 December, 2016.

[7] Goodspeed, T. (2009). Extracting keys from second generation zigbee chips. Black Hat USA, 9.

[8] Goodspeed, T. (2016). Sourceforgenet. Retrieved 15 December, 2016, from http://goodfet.sourceforge.net/

[9] Goodspeed, T., Bratus, S., Melgares, R., Speers, R., & Smith, S. W. (2012, January). Api-do: Tools for exploring the wireless attack surface in smart meters. In System Science (HICSS), 2012 45th Hawaii International Conference on (pp. 2133-2140). IEEE.

[10] Harari, E. (n.d.). OpenSSHd 7.2p2 - Username Enumeration (PoC). Retrieved March 12, 2017, from https://www.exploit-db.com/exploits/40113/

[11] Home Page of EU GDPR. (n.d.). Retrieved July 27, 2017, from http://www.eugdpr.org/

[12] Hunt, T. (n.d.). Automated Security Analyser for ASP.NET Websites. Retrieved March 2, 2017, from https://asafaweb.com/

[13] Jun, L. (2016). Defconorg. Retrieved 15 December, 2016, from https://media.defcon.org/DEF CON 23/DEF CON 23 presentations/DEFCON-23-Li-Jun-Yang-Qing-I-AM-A-NEWBIE-YET-I-CAN-HACK-ZIGBEE.pdf

[14] Jun, L. (2016). YouTube. Retrieved 15 December, 2016, from https://www.youtube.com/watch?v=xgNT05l6Jlw

[15] Keep Calm and Implement ZigBee Security. (n.d.). Retrieved July 16, 2017, from http://ioticity.solutions/news/54

[16] Kenkeiras. Vulnerability & Exploit Database. (n.d.). Retrieved May 10, 2017, from https://www.rapid7.com/db/modules/auxiliary/scanner/ssh/ssh_enumusers

[17] Lee, T. (2016). SecuritySynapse. Retrieved November 16, 2016, from http://securitysynapse.blogspot.pt/2015/12/fun-with-zigbee-wireless-part-iv.html

[18] Lee, T. (2016). SecuritySynapse. Retrieved November 16, 2016, from http://securitysynapse.blogspot.pt/2015/12/fun-with-zigbee-wireless-part-v.html

[19] Lifchitz, R. (Writer). (2016, July 2). ZigBee security review of a famous French set-top box. Live performance in Paris.

[20] Low Power Wireless and ZigBee Networking Workshop - ZigBee Stack . (n.d.). Retrieved June 28, from http://processors.wiki.ti.com/images/8/8a/08_-_ZigBee_Stack.pdf

[21] Netcraft Extension. (n.d.). Retrieved March 2, 2017, from http://toolbar.netcraft.com/

[22] Norton (n.d.). IoT. Retrieved July 27, 2017, from https://us.norton.com/internetsecurity-iot-5-predictions-for-the-future-of-iot.html

[23] Offensive Security's Exploit Database Archive. (n.d.). Retrieved Spring, 2017, from https://www.exploit-db.com/

[24] Olawumi, O., Haataja, K., Asikainen, M., Vidgren, N., & Toivanen, P. (2014). Three practical attacks against ZigBee security: Attack scenario definitions, practical experiments, countermeasures, and lessons learned. 2014 14th International Conference on Hybrid Intelligent Systems. doi:10.1109/his.2014.7086198

[25] Openbsd » Openssh » 6.1 : Security Vulnerabilities. (n.d.). Retrieved January 6, 2017, from https://www.cvedetails.com/vulnerability-list/vendor_id-97/product_id-585/version_id-156381/Openbsd-Openssh-6.1.html

[26] OpenSSH 7.2p2 - Username Enumeration. (2015) Retrieved March 12, 2017, from https://www.exploit-db.com/exploits/40136/

[27] Paramount digital marketing. (2016). Willowcouk. Retrieved 15 December, 2016, from http://www.willow.co.uk/html/telosb_mote_platform.php

[28] Phorn, J. (n.d.). GitHub - jeanphorn/wordlist: Collection of some common. Retrieved December 16, 2017, from https://github.com/jeanphorn/wordlist

[29] Razouk, W., Crosby, G. V., & Sekkaki, A. (2014). New Security Approach for ZigBee Weaknesses. Procedia Computer Science, 37, 376-381. doi:10.1016/j.procs.2014.08.056

[30] Retrieved October 21, 2016, from https://www.elprocus.com/wp-content/uploads/2014/05/34.jpg

[31] Richardson, T. (n.d.). ZigBee, the IoT, and Global Growth. Retrieved June 9, 2017, from http://www.iot-today.com/main/articles/zigbee-the-iot-and-global-growth/

[32] Riverloopsec. (2016). Api-Mote. Retrieved 15 December, 2016, from http://informatik.hs-furtwangen.de/~pip/appendix/

[33] Riverloopsec. (2016). GitHub - KillerBee. Retrieved 15 December, 2016, from https://github.com/riverloopsec/killerbee

[34] Ryan Speers. (2016). Rmspeerscom. Retrieved 15 December, 2016, from http://rmspeers.com/archives/274

[35] SecurityFocus. (n.d.). Retrieved Spring, 2017, from http://www.securityfocus.com/

[36] Silva, R., & Nunes, S. (2006). Security in IEEE 80.15.4 Standard. Retrieved December 8, from https://web.archive.org/web/20080315160351/http://www.estig.ipbeja.pt/~rmss/uploads/materaJan06-3.pdf

[37] Silverman, C. (n.d.). 7 Creepy Baby Monitor Stories That Will Terrify All Parents. Retrieved July 27, 2017, from https://www.buzzfeed.com/craigsilverman/creeps-hack-baby-monitors-and-say-terrifying-thing?utm_term=.rt9amnyJL#.xs2ZvyQwL

[38] Snyder, B. (2016, May 25). Study finds Google, Twitter and Facebook keep closest tabs on users. Retrieved July 27, 2017, from http://www.cio.com/article/3074340/privacy/google-twitter-facebook-cookies-user-tracking.html

[39] Speers, R. (2011). IEEE 802.15. 4 Wireless Security: Self-Assessment Frameworks.

[40] Sr. Computador - Reparacao e manutencao de computadores. (n.d.). Retrieved June 5, 2017, from http://www.senhorcomputador.pt/

[41] Testa, J. (2017, July 06). Jtesta/ssh-mitm. Retrieved July 10, 2017, from https://github.com/jtesta/ssh-mitm

[42] Vidgren, N., Haataja, K., Patino-Andres, J. L., Ramirez-Sanchis, J. J., & Toivanen, P. (2013). Security Threats in ZigBee-Enabled Systems: Vulnerability Evaluation, Practical Experiments, Countermeasures, and Lessons Learned. 2013 46th Hawaii International Conference on System Sciences. doi:10.1109/hicss.2013.475

[43] Vulnerability Details : CVE-2016-8858. (n.d.). Retrieved January 6, 2017, from https://www.cvedetails.com/cve/CVE-2016-8858/

[44] Vulnerability & Exploit Database. (n.d.). Retrieved June 20, 2017, from https://www.rapid7.com/db/vulnerabilities/openbsd-openssh-cve-2006-5229

[45] Web Server Security Test | High-Tech Bridge. (n.d.). Retrieved March 2, 2017, from https://www.htbridge.com/websec/

[46] Wright, J. (2016). Willhackforsushicom. Retrieved 15 December, 2016, from http://www.willhackforsushi.com/presentations/toorcon11-wright.pdf

[47] Wright, J. (2016). YouTube. Retrieved 15 December, 2016, from https://www.youtube.com/watch?v=BkVcElfOVyw

[48] Wright, J & Cache, J. (2015). Hacking Exposed Wireless. (3rd ed.). : Mc Graw Hill Education.

[49] Yang, B. (2009, December). Study on security of wireless sensor network based on ZigBee standard. In Computational Intelligence and Security, 2009. CIS'09. International Conference on (Vol. 2, pp. 426-430). IEEE.

[50] ZigBee 3.0 – Facilitating the Internet of Things. (2016) Retrieved June 20, 2017, from http://www.nxp.com/docs/en/brochure/75017677.pdf

[51] Zigbee alliance. (2014). ZigBee Specification. USA: ZigBee Standards Organization.

[52] Zigbee light link master key • r/hackernews. (n.d.). Retrieved July 8, 2017, from https://www.reddit.com/r/hackernews/comments/2zzt2x/zigbee_light_link_master_key/

[53] ZigBee, M. (n.d.). Retrieved July 10, 2017, from https://mobile.twitter.com/MayaZigBee/status/579723961661022209

[54] ZigBee network topologies: star, cluster tree and mesh. | Open-i. (n.d.). Retrieved October 21, 2016, from https://openi.nlm.nih.gov/detailedresult.php?img=PMC3675532_sensors-08-03067f2&req=4

[55] Zillner, T. (2015). ZIGBEE EXPLOITED The good, the bad and the ugly. Retrieved May 01, 2017, from https://www.blackhat.com/docs/us-15/materials/us-15-Zillner-ZigBee-Exploited-The-Good-The-Bad-And-The-Ugly-wp.pdf

ANEXOS

ANEXO I
Processo de instalação de *firmware* KillerBee através da Atmel-ICE para RZUSBstick

Processo de instalação de *firmware* KillerBee através da Atmel-ICE para RZUSBstick

O processo de instalação do *firmware* KillerBee usando a programadora JTAG Atmel-ICE deve seguir os seguintes passo:

1. Instalar o AtmelStudio e executar o programa
2. Ligar Atmel-ICE a uma porta USB
3. Ligar o conetor de 10 pinos ao RZUSBstick e a outra ponta à *slot* AVR
4. Ligar RZUSBstick a uma porta USB
5. No AtmelStudio ir a *Tools > Device Programming*
6. Escolher as seguintes opções:
    - *Tools*: Atmel-ICE
    - *Device*: AT90USB1287
    - *Interface*: JTAG
7. Pressionar *Apply*
8. No *Device Signature*, pressionar *Read*. Caso devolva algum erro, é necessário rodar o conetor de 10 pinos no RZUSBstick
9. Escolher *Memories*
10. No menu flash pressionar *Read* e guardar um *backup* do *firmware*
11. Escolher o ficheiro com o nome *firmware* e selecionar *Erase device before programming*, e também a opção *Verify flash after programming*
12. Pressionar *Program* e deve aparecer na caixa de mensagens:
    - *Erasing device*…. OK
    - *Programming Flash*… OK
    - *Verifying Flash*… OK